FICHE SYNTHÈSE : ADRESSES IPV4

Définition

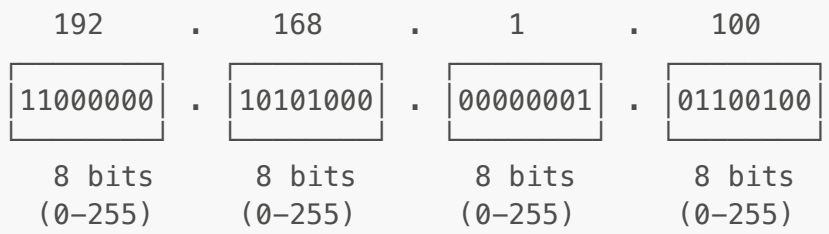
IPv4 (Internet Protocol version 4) = Adresse logique identifiant de manière unique un équipement sur un réseau IP.

- **Couche OSI** : Couche 3 (Réseau)
- **Portée** : Locale ou Internet (routable)
- **Taille** : 32 bits (4 octets)
- **Format** : Notation décimale pointée
- **Nombre total** : 2^{32} = 4 294 967 296 adresses possibles

---

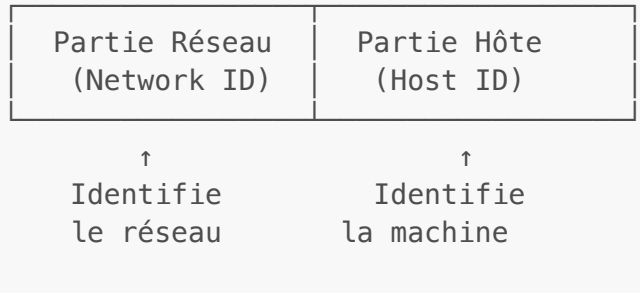
Structure de l'Adresse IPv4

Format standard : 4 octets = 32 bits



Composition

Une adresse IPv4 se compose de deux parties :



La séparation est définie par le **masque de sous-réseau**.

Classes d'Adresses (Classful)

Système historique (obsolète depuis CIDR, mais toujours enseigné).

Classe	Premier octet	Plage	Masque par défaut	Nb réseaux	Nb hôtes/réseau
A	0-127	0.0.0.0 - 127.255.255.255	/8 (255.0.0.0)	128	16 777 214
B	128-191	128.0.0.0 - 191.255.255.255	/16 (255.255.0.0)	16 384	65 534
C	192-223	192.0.0.0 - 223.255.255.255	/24 (255.255.255.0)	2 097 152	254
D	224-239	224.0.0.0 - 239.255.255.255	N/A	Multicast	-
E	240-255	240.0.0.0 - 255.255.255.255	N/A	Réservé (expérimental)	-

Identification des bits de début

```
Classe A : 0xxxxxxx.xxxxxxxx.xxxxxxxx.xxxxxxxx
Classe B : 10xxxxxx.xxxxxxxx.xxxxxxxx.xxxxxxxx
Classe C : 110xxxxx.xxxxxxxx.xxxxxxxx.xxxxxxxx
Classe D : 1110xxxx.xxxxxxxx.xxxxxxxx.xxxxxxxx
Classe E : 1111xxxx.xxxxxxxx.xxxxxxxx.xxxxxxxx
```

Masques de Sous-Réseau

Notation CIDR (Classless Inter-Domain Routing)

```
192.168.1.0/24
            └─ Nombre de bits à 1 dans le masque (préfixe)
```

Table de conversion CIDR

CIDR	Masque décimal	Nombre d'hôtes	Cas d'usage
/8	255.0.0.0	16 777 214	Très grand réseau
/16	255.255.0.0	65 534	Grand réseau
/24	255.255.255.0	254	Réseau standard (LAN)

CIDR	Masque décimal	Nombre d'hôtes	Cas d'usage
/25	255.255.255.128	126	Sous-réseau moyen
/26	255.255.255.192	62	Petit sous-réseau
/27	255.255.255.224	30	Très petit réseau
/28	255.255.255.240	14	Micro-réseau
/29	255.255.255.248	6	Lien point-à-point
/30	255.255.255.252	2	Lien entre routeurs
/31	255.255.255.254	2	Lien P2P (RFC 3021)
/32	255.255.255.255	1	Hôte unique

Formules de calcul

```
Nombre d'hôtes = 2^(32 – préfixe) – 2

Exemples :
/24 → 2^(32–24) – 2 = 2^8 – 2 = 254 hôtes
/26 → 2^(32–26) – 2 = 2^6 – 2 = 62 hôtes
```

Pourquoi -2 ?

- 1 adresse pour le **réseau** (tous les bits hôte à 0)
- 1 adresse pour le **broadcast** (tous les bits hôte à 1)

Adresses Spéciales

1. Adresses Privées (RFC 1918)

Non routables sur Internet, utilisées dans les réseaux locaux.

Classe	Plage	CIDR	Nombre de réseaux
A	10.0.0.0 - 10.255.255.255	10.0.0.0/8	1 réseau de classe A
B	172.16.0.0 - 172.31.255.255	172.16.0.0/12	16 réseaux de classe B
C	192.168.0.0 - 192.168.255.255	192.168.0.0/16	256 réseaux de classe C

2. Adresse de Loopback

```
127.0.0.0/8 (127.0.0.1 – 127.255.255.255)
```

- **127.0.0.1** : Interface locale (localhost)

- Test de la pile TCP/IP sans carte réseau
- Trafic ne quitte jamais la machine

3. APIPA (Automatic Private IP Addressing)

```
169.254.0.0/16 (169.254.0.1 – 169.254.255.254)
```

- Auto-configuration en l'absence de DHCP
- Windows, macOS, Linux
- Non routable

4. Adresse de Broadcast

```
255.255.255.255
```

- Broadcast limité (limited broadcast)
- Envoi à toutes les machines du segment local
- Ne traverse pas les routeurs

5. Adresse Multicast (Classe D)

```
224.0.0.0 – 239.255.255.255
```

Exemples :

- **224.0.0.1** : Tous les hôtes du segment
- **224.0.0.2** : Tous les routeurs du segment
- **224.0.0.5** : OSPF (protocole de routage)
- **239.255.255.250** : SSDP (UPnP)

6. Autres Adresses Réservées

Plage	Usage
0.0.0.0/8	Réseau actuel (source uniquement)
0.0.0.0	Adresse par défaut / aucune adresse
192.0.2.0/24	Documentation (TEST-NET-1)
198.51.100.0/24	Documentation (TEST-NET-2)
203.0.113.0/24	Documentation (TEST-NET-3)
240.0.0.0/4	Réservé (Classe E)

Calculs de Sous-Réseaux

Exemple 1 : Réseau 192.168.1.0/24

```
Adresse réseau :   192.168.1.0
Masque :           255.255.255.0 (/24)
Première IP :      192.168.1.1
Dernière IP :      192.168.1.254
Broadcast :        192.168.1.255
Nombre d'hôtes :   254
```

Exemple 2 : Découper 192.168.1.0/24 en 4 sous-réseaux

```
Masque d'origine : /24 (254 hôtes)
Nouveau masque :   /26 (62 hôtes par sous-réseau)

Sous-réseau 1 :    192.168.1.0/26    (1-62)
Sous-réseau 2 :    192.168.1.64/26   (65-126)
Sous-réseau 3 :    192.168.1.128/26  (129-190)
Sous-réseau 4 :    192.168.1.192/26  (193-254)
```

Méthode de calcul rapide

Incrément de sous-réseau = 256 - valeur du dernier octet du masque

Exemple avec /26 (255.255.255.192) :

```
256 - 192 = 64
Donc : 0, 64, 128, 192 (4 sous-réseaux)
```

Trouver l'adresse réseau d'une IP

Méthode : ET logique entre l'IP et le masque

```
IP :       192.168.1.75
Masque :   255.255.255.192 (/26)

Binaire :
IP :       11000000.10101000.00000001.01001011
Masque :   11111111.11111111.11111111.11000000
           ----------------------------------
Réseau :   11000000.10101000.00000001.01000000
           = 192.168.1.64/26
```

Commandes Pratiques

Linux

```
# Afficher la configuration IP
ip addr show
ip a

# Interface spécifique
ip addr show eth0

# Afficher les routes
ip route show
ip route get 8.8.8.8

# Ancienne méthode (ifconfig)
ifconfig
ifconfig eth0

# Configurer une IP statique (temporaire)
sudo ip addr add 192.168.1.100/24 dev eth0
sudo ip link set eth0 up

# Supprimer une IP
sudo ip addr del 192.168.1.100/24 dev eth0

# Tester la connectivité
ping -c 4 192.168.1.1
traceroute 8.8.8.8
```

Windows

```
# Afficher la configuration
ipconfig
ipconfig /all

# Libérer/renouveler IP DHCP
ipconfig /release
ipconfig /renew

# Afficher la table de routage
route print

# Ping et traceroute
ping 192.168.1.1
tracert 8.8.8.8

# PowerShell
Get-NetIPAddress
```

```
Get-NetRoute
Test-NetConnection -ComputerName 192.168.1.1
```

macOS

```
# Afficher la configuration
ifconfig
networksetup -listallnetworkservices
networksetup -getinfo "Wi-Fi"

# Renouveler IP DHCP
sudo ipconfig set en0 DHCP
```

Calculateurs de sous-réseaux en ligne de commande

```
# ipcalc (Linux)
ipcalc 192.168.1.0/24

# sipcalc
sipcalc 192.168.1.0/24

# Python one-liner
python3 -c "import ipaddress; net =
ipaddress.IPv4Network('192.168.1.0/24'); print(f'Réseau:
{net.network_address}\nBroadcast: {net.broadcast_address}\nHôtes:
{net.num_addresses-2}')"
```

Aspects Sécurité

1. IP Spoofing (Usurpation d'IP)

Principe : Falsifier l'adresse IP source dans les paquets pour :

- Cacher son identité
- Contourner des filtres basés sur IP
- Attaques DDoS (réflexion/amplification)

Contre-mesures :

- Filtrage anti-spoofing (ingress/egress filtering)
- Reverse Path Forwarding (RPF)
- Authentification forte (pas seulement basée sur IP)

2. Scanning de Réseaux

```
# Nmap : scanner un réseau
nmap -sn 192.168.1.0/24    # Ping scan
nmap -p- 192.168.1.10      # Tous les ports

# Masscan : scanner rapide
masscan 192.168.1.0/24 -p80,443

# fping : ping multiple
fping -g 192.168.1.0/24
```

3. Reconnaissance Passive

```
# Whois : information sur IP publique
whois 8.8.8.8

# DNS inverse
dig -x 8.8.8.8
nslookup 8.8.8.8
```

4. Attaques par Fragmentation IP

- **Ping of Death** : Paquets ICMP fragmentés trop grands
- **Teardrop** : Fragments IP qui se chevauchent

Protection : Pare-feu, détection d'intrusion (IDS)

NAT (Network Address Translation)

Principe

Permet à plusieurs machines privées de partager une IP publique.

```
Réseau privé                  NAT                    Internet
192.168.1.10:5000  ————————>  [Routeur]  ————————>  203.0.113.5:5000
192.168.1.20:5001  ————————>  203.0.113.5:5001      (IP publique)
192.168.1.30:5002  ————————>  203.0.113.5:5002
```

Types de NAT

Type	Description
SNAT	Source NAT (IP privée → IP publique)
DNAT	Destination NAT (redirection de ports)
PAT	Port Address Translation (NAT + ports)

Type	Description
Full Cone NAT	Restriction minimale
Restricted NAT	Filtrage par IP
Port Restricted	Filtrage par IP:port
Symmetric NAT	Mapping unique par destination

Script Python : Manipulation IPv4

```python
import ipaddress

def analyser_ip(ip_avec_masque):
    """Analyser une adresse IP avec son masque"""
    network = ipaddress.IPv4Network(ip_avec_masque, strict=False)

    print(f"Adresse IP : {ip_avec_masque}")
    print(f"Adresse réseau : {network.network_address}")
    print(f"Masque : {network.netmask} (/{network.prefixlen})")
    print(f"Broadcast : {network.broadcast_address}")
    print(f"Première IP : {network.network_address + 1}")
    print(f"Dernière IP : {network.broadcast_address - 1}")
    print(f"Nombre d'hôtes : {network.num_addresses - 2}")
    print(f"Adresse privée : {network.is_private}")

def ip_dans_reseau(ip, reseau):
    """Vérifier si une IP appartient à un réseau"""
    ip_obj = ipaddress.IPv4Address(ip)
    net_obj = ipaddress.IPv4Network(reseau, strict=False)
    return ip_obj in net_obj

def decoupe_reseau(reseau, nb_sous_reseaux):
    """Découper un réseau en sous-réseaux"""
    network = ipaddress.IPv4Network(reseau, strict=False)

    # Calculer le nouveau préfixe
    import math
    bits_necessaires = math.ceil(math.log2(nb_sous_reseaux))
    nouveau_prefixe = network.prefixlen + bits_necessaires

    sous_reseaux = list(network.subnets(new_prefix=nouveau_prefixe))

    return sous_reseaux[:nb_sous_reseaux]

# Exemples d'utilisation
print("=== Analyse d'une IP ===")
analyser_ip("192.168.1.100/24")

print("\n=== Test d'appartenance ===")
print(ip_dans_reseau("192.168.1.50", "192.168.1.0/24"))  # True
```

```
print(ip_dans_reseau("192.168.2.50", "192.168.1.0/24"))  # False

print("\n=== Découpage en sous-réseaux ===")
sous_reseaux = decoupe_reseau("192.168.1.0/24", 4)
for i, sr in enumerate(sous_reseaux, 1):
    print(f"Sous-réseau {i} : {sr}")
```

Script Bash : Calculateur Subnet

```
#!/bin/bash

# Fonction pour calculer les infos d'un réseau
subnet_calc() {
    local ip=$1
    local cidr=$2

    # Utiliser ipcalc si disponible
    if command -v ipcalc &> /dev/null; then
        ipcalc -b -n -m $ip/$cidr
    else
        # Alternative avec Python
        python3 -c "
import ipaddress
net = ipaddress.IPv4Network('$ip/$cidr', strict=False)
print(f'Network: {net.network_address}')
print(f'Netmask: {net.netmask}')
print(f'Broadcast: {net.broadcast_address}')
print(f'HostMin: {net.network_address + 1}')
print(f'HostMax: {net.broadcast_address - 1}')
print(f'Hosts: {net.num_addresses - 2}')
        "
    fi
}

# Utilisation
subnet_calc 192.168.1.0 24
```

IPv4 vs IPv6

Caractéristique	IPv4	IPv6
Taille	32 bits	128 bits
Format	Décimal pointé	Hexadécimal
Adresses totales	~4,3 milliards	340 undécillions
Broadcast	Oui	Non (multicast)

Caractéristique	IPv4	IPv6
NAT	Nécessaire	Optionnel
Fragmentation	Routeurs	Hôtes uniquement
Configuration	DHCP	SLAAC / DHCPv6
En-tête	Variable (20-60 octets)	Fixe (40 octets)

Points Clés à Retenir

✅ **32 bits** = 4 octets en notation décimale pointée
✅ **Partie réseau + partie hôte** définie par le masque
✅ **CIDR** : notation moderne avec /préfixe
✅ **Adresses privées** : 10.x, 172.16-31.x, 192.168.x
✅ **Loopback** : 127.0.0.1
✅ **Broadcast** : 255.255.255.255
✅ **NAT** : permet le partage d'IP publique
✅ Calcul d'hôtes : **2^(32-préfixe) - 2**

Exercices Pratiques

Exercice 1 : Identification

Pour chaque adresse, identifier : classe, type (privée/publique), usage

- 10.0.0.1
- 172.217.16.142
- 192.168.1.1
- 127.0.0.1
- 224.0.0.1

Exercice 2 : Calcul de sous-réseaux

Découper 192.168.10.0/24 en 8 sous-réseaux égaux. Donner pour chacun :

- Adresse réseau
- Première IP utilisable
- Dernière IP utilisable
- Broadcast

Exercice 3 : Détermination du masque

Combien de bits faut-il pour avoir :

- 50 hôtes minimum ?
- 500 hôtes minimum ?
- 5000 hôtes minimum ?

Ressources

- **RFC 791** : Internet Protocol
- **RFC 1918** : Address Allocation for Private Internets
- **RFC 3927** : Dynamic Configuration of IPv4 Link-Local Addresses
- **Calculateurs en ligne** :
  - https://www.subnet-calculator.com/
  - https://www.calculator.net/ip-subnet-calculator.html
- **Outil** : `ipcalc`, `sipcalc` (Linux)